

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Visual Analogies at Multiple Levels of Abstraction

Permalink

<https://escholarship.org/uc/item/1kf1x6xz>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 29(29)

ISSN

1069-7977

Authors

Yaner, Patrick W.
Goel, Ashok K.

Publication Date

2007

Peer reviewed

Visual Analogies at Multiple Levels of Abstraction

Patrick W. Yaner (yaner@cc.gatech.edu)

Ashok K. Goel (goel@cc.gatech.edu)

Design Intelligence Laboratory

School of Interactive Computing, Georgia Institute of Technology
Atlanta, GA 30332-0760

Abstract

We describe a method for constructing a teleological model of an unlabelled 2D line drawing by analogy to a known model of a drawing with similar structure. The source case is represented as a schema that contains its line drawing and its teleological model represented at multiple levels of abstraction: the lines and intersections in the drawing, the shapes, the structural components and connections, the causal interactions and processes, and the function of the device depicted in the drawing. Given a target drawing and a relevant source case, our method of *compositional analogy* first constructs a representation of the lines and the intersections in the target drawing, then uses the mappings at the level of line intersections to transfer the shape representations from the source case to the target, next uses the mappings at the level of shapes to transfer the full teleological model of the depicted system from the source to the target.

Keywords: diagrammatic reasoning; analogy; visual reasoning; case-based reasoning; design

Motivation and Goals

Humans often communicate with drawings, and human designers are able to read and understand drawings of designs. When shown a drawing of a new device, an expert designer in the device domain may not only identify device components but also explain what the device does and how it works. We have developed a theory of how this might be done: by transfer of a teleological model of a known drawing with a similar or related structure. This theory is implemented in a program called *Archytas*. This program reads in a 2D unlabelled drawing and, given the source drawing and associated teleological model, attempts to infer by analogy, (1) a representation of the shapes and spatial relations in the target drawing, (2) a representation of the structural components and connections of the device depicted in the drawing, (3) a qualitative representation of the causal interactions and processes in the device, and (4) a representation of the function of the device. This method of *compositional analogy* works iteratively to successively higher levels of abstraction, interleaving mapping and transfer at various levels to construct a new teleological model.

To see the need for mapping and transfer at multiple abstraction levels, let us consider the task of mapping the source drawing illustrated in figure 1(a) to the similar target drawing illustrated in figure 1(b). If we treat the problem as one of first recognizing the geometric elements and spatial relations among them, then we can treat this representation as a labelled graph: *A* contains *B*, *C* is adjacent to *D*, and so on. A graph-theoretic method for analogy-based recognition may

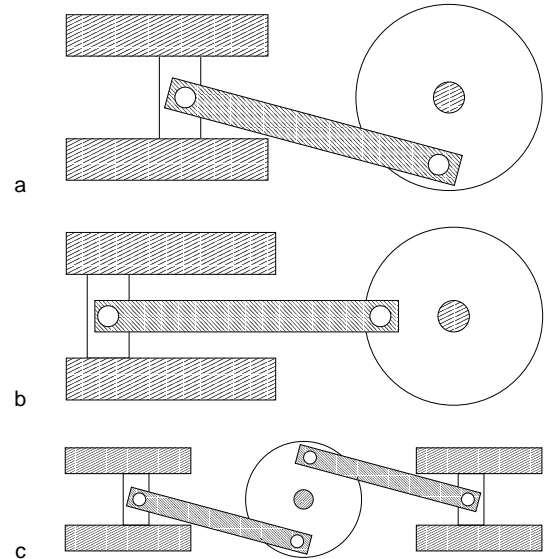


Figure 1: (a) A sample *source* drawing of a piston and crankshaft assembly. (b) A *target* drawing of the same device with the piston now at the top of its range of motion, and (c) another sample *target* drawing of a two piston and crankshaft assembly. The shape patterns as well as the SBF model is transferred from the source (a) to an input drawing such as (b) or (c).

then be used to find a consistent mapping between the graphs representing the source and target drawings. However, the method runs into difficulty for the target drawing shown in figure 1(c) with figure 1(a) as the source drawing. In this problem, the *number* of components, and thus the number of shapes, is different, and either the graph-theoretic method would have to relax the constraint of one-to-one mapping, or else the analogy would have to be performed twice in order to transfer a model successfully from figure 1(a) to figure 1(c). Figure 2 illustrates a similar example from the domain of door latches.

To address the above difficulties, our method of *compositional analogy* performs analogy at *multiple levels of abstraction*. The analogical mapping and transfer is enabled by organizing knowledge of the source case at multiple abstraction levels. Figure 3 illustrates the knowledge organization in a source case. The function, behavior, structure, shape, and drawing in a source case form an abstraction hi-

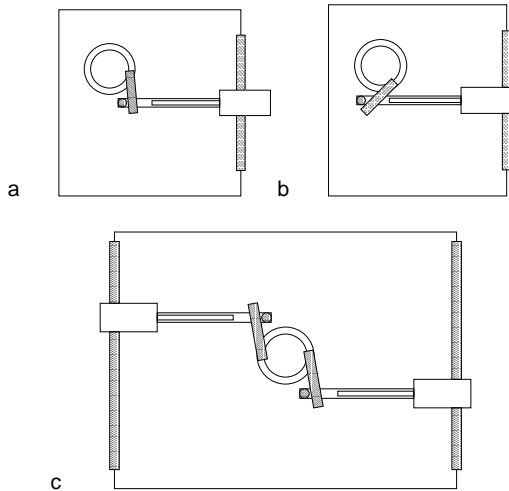


Figure 2: (a) A sample *source* drawing of a door latch mechanism. (b) A *target* drawing of the same device with the door latch now pulled inward, and (c) another sample *target* drawing of a double door latch (a latch on each side). Again the shape patterns as well as the SBF model is transferred from the source (a) to an input drawing such as (b) or (c).

erarchy called a DSSBF (Drawing Shape Structure Behavior Function) model. Structure in this hierarchy is a specification of components and structural relations, along with properties (height, width, etc.) and variable parameters (e.g. position, angle) of each. Behavior is a qualitative representation of discrete states and transitions, where transitions are annotated with the causes of that state transition. Function is an abstraction of behavior, the purpose for which the device is intended, and is represented as an input-output schema. These DSSBF models extend our earlier work on Structure-Behavior-Function (SBF) models (Goel & Chandrasekaran, 1989; Goel, 1991).

Our method of compositional analogy first gathers individual lines and circles and intersection points in the target drawing into shapes, and then finds mappings between the source and the target at this level of intersections. Then, it groups these mappings and transfers shapes from the source to the target. Next, it finds a mapping between the source and the target at the shape level, and then transfers the structural model. This transfer process produces a mapping at the level of components and connections among them, and the method uses this structure-level mapping to transfer the behaviors from source to target. Finally, the method uses the behavioral-level mapping to construct a functional representation of the depicted device.

The Structure of Shapes

When a drawing is represented symbolically for analogical comparison, symbols typically are associated with shapes. In this symbolic representation, the shape, which is a composite geometric structure (a point set in the real plane), becomes an atomic entity. If analogy involves the alignment of symbol structures, then analogical comparison between two drawings can only be successful when these symbol structures are iso-

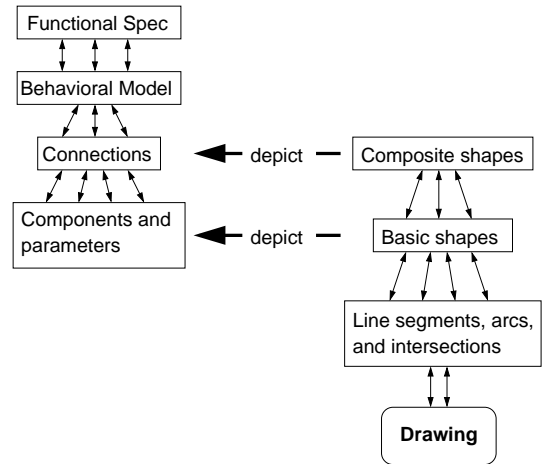


Figure 3: The multi-level hierarchy of drawing, basic shapes, and composite shapes, related to the teleological model on the left with its components and connections between components. Basic shapes depict components, and composite shapes, made up of basic shapes, depict connections between components.

morphic. However, whether or not two symbol structures are aligned depends on how the shapes in the structures are decomposed. Figure 4 shows three structurally incompatible ways of representing even a very basic shape. To address this issue, we represent shapes in the source case at multiple levels of abstraction, and use the shape representation in the source to help resolve ambiguities in constructing a representation of shapes in the target.

Basic and Composite Shapes

The first step of the process is to match each shape in the source to the target drawing. We need a robust and canonical representation of shapes, one that is robust in face of the *visual binding problem*: line segments should match regardless of whether they are drawn in the vector graphics file as a single vector or as multiple vectors. For instance, the rectangle corresponding to the upper half of the cylinder in figure 1(a) might be drawn as four perpendicular lines, or it might be *six* lines (if the edge touching the piston is three segments instead of one); the shape must match either way. This is achieved using what we call the augmented line intersection graph of the drawing, shown in figure 5 (Yaner & Goel, 2007).

The line intersection graph represents the most basic topological information in the drawing. To reduce the search space for mapping, Archytas augments this graph with additional spatial information. Firstly, since each intersection is a pair of line (or arc) sets, Archytas adds a flag on each edge (intersection) indicating perpendicularity. This prevents Archytas from matching, say, a rectangle to a rhombus. Secondly, each topological face in the drawing bounded by a cycle of line and arc sets is represented as a special vertex linked to the points and lines on its boundary. This represents the planar dual of the input drawing (regarded as a plane graph). Archytas represents the basic elements of a 2D line drawing (line segments, circles and circular arcs) using this augmented

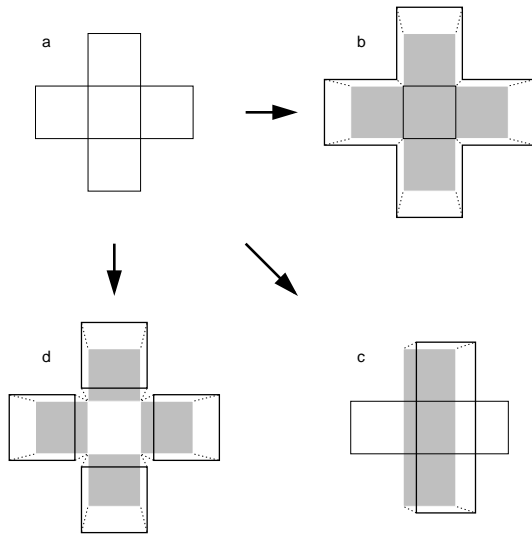


Figure 4: Three structurally inconsistent ways of decomposing (a) a simple compound cross shape: (b) an outline cross shape with an inner square, (c) a pair of overlapping rectangles, or even (d) four squares touching at the corners.

line intersection graph, which takes lines and arcs and circles as nodes and intersections between them as labelled arcs.

Depending on the perspective of the drawing, a particular component may be depicted by an actual shape, regarded as a set of connected lines or arcs, or not at all. When a component is depicted by a shape, which is a *subgraph* of the intersection graph. Each component in the model shown in figure 6 is linked to a subgraph of the intersection graph called a “basic shape”. A connection between components in the model is a union of several entities, and so each connection is linked to a subgraph called a “composite shape.” A composite shape is a union of two or more “basic shapes”, forming a larger subgraph of the intersection graph. This forms the right half of the multi-level hierarchy shown in figure 3.

Analogical Mapping and Transfer

The goal of the mapping stage is to align shapes in the representations of the source and target drawings by finding correspondences between the basic and composite shapes in their augmented line intersection graphs. In particular, Archytas attempts to find mappings from each basic and composite shape to some subgraph of the target drawing’s line intersection graph. Here the mapping must be exact: each element of the shape description must match, and so the algorithm is one of *subgraph isomorphism*.

The source drawing is divided into basic and composite shapes. The drawing in figure 1(a), for instance, has the following depicted components, each of which has one basic shape associated with it: (i) piston, (ii) connecting rod, (iii) crankshaft, and (iv) cylinder. In addition, the following structural relations are shown in the drawing, each of which has a composite shape associated with it: (i) cylindrical joint between piston and cylinder, (ii) revolute joint between piston and connecting rod, and (iii) revolute joint between connecting rod and crankshaft. Thus there are three composite shapes

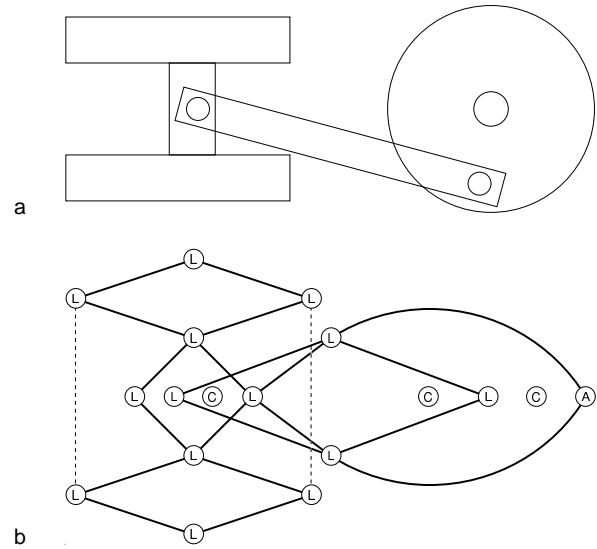


Figure 5: An example of (a) a line drawing of a piston and crankshaft with fill patterns and layering ignored (as they are in Archytas), and (b) its corresponding line intersection graph representation: L stand for for line, C for circle, A for arc (edge labels have been omitted for clarity). In addition, each bounded region (or *face*) is represented as a separate node (not shown) linked to the vertices and edges on its boundary. The dotted lines indicate collinear but disconnected lines.

that overlap with four basic shapes. The crankcase and it’s structural relations, recall, are undepicted.

In order to match these shapes to either figure 1(b) or figure 1(c), Archytas first begins by trying to compute a subgraph isomorphism from *each* composite shape to the target intersection graph. Upon finding these, Archytas breaks the composite shape mappings into corresponding basic shape mappings, so that when two composites with a basic shape in common overlap in the mapping as well, that basic shape can be inferred of the target.

Symmetric Mappings

In figure 1(a), the composite shape corresponding to the piston-cylinder connection has four potential mappings onto the target 1(b). The piston/connecting-rod connection has two rectangles and some circles as its constituents, and there are again four mappings to the target, but not each of them overlaps properly with each mapping for the piston-cylinder shape. Thus, if we chose the wrong pair, we might get *two* pistons instead of one, with one connected to the connecting rod, and the other connected to the cylinder. This is clearly an error of reasoning.

Note that the various mappings of a single shape in the target are *symmetric*. To see this, let m_1, m_2 be two mappings for a shape. In general, m_1 and m_2 may map the source shape onto different areas of the target, but if they do not, they still may be incompatible. For instance, if a, b, c are the corners of a triangle in the source, and x, y, z of a similar one in the

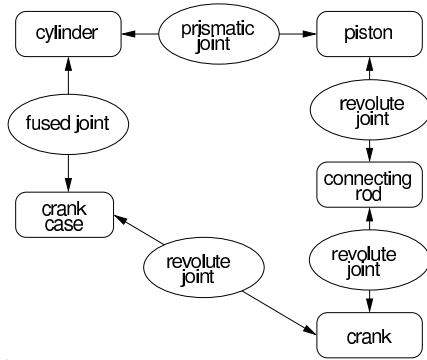


Figure 6: An illustration of the structural model showing components (boxes) and structural relations (ovals). Properties and variable parameters of components are specified in slot values for each component frame.

target, we might have:

$$\begin{array}{ll}
 m_1 : a \mapsto x & m_2 : a \mapsto z \\
 m_1 : b \mapsto y & m_2 : b \mapsto y \\
 m_1 : c \mapsto z & m_2 : c \mapsto x
 \end{array}$$

These two mappings are symmetric and therefore equivalent: both are mapping the same shape to the same points in the target but in different orientations. This is important from the perspective of transfer: the target shape will be identical with both mappings, so they are effectively equivalent.

We can then *group* symmetric mappings into sets. Archytas computes *all* mappings of each composite shape in the target, and divide them into these sets of symmetric mappings. For each such set, the composite shape mappings are divided into basic shape mappings and these two are grouped using symmetry. Archytas then transfers *one* composite shape in the target for *each group* of composite shape mappings, and *one* basic shape for *each group* of basic shape mappings. The algorithms are presented in the next section.

Shape Mapping and Transfer

The goal of the shape transfer algorithm is to bring structure to the target drawing: that is, to use patterns to divide the lines and intersections into basic and composite shapes, which inform Archytas of visual patterns depicting components and structural relations. It is important to compute *all* the mappings of a given shape so that all the relationships can be found. Each mapping group informs a new shape in the target. The result is a *shape-level mapping*: a mapping at the level of whole shapes rather than individual lines and intersection points. From here, the transfer of structural elements—components and structural relations—can take place. This process is illustrated in figure 7.

The algorithm treats shape matching as one of satisfying constraints. Each composite shape in the source is treated as a pattern, and the elements of the intersection graph in the target are matched to that pattern. Archytas attempts to find all consistent ways of matching a given pattern to a drawing, grouping symmetric ones as discussed above, where *symmet-*

ric means that the set of points being mapped are identical in both, even if the details of the mapping are not.

To constrain potential mappings, when two lines or intersection points are both on the boundary of a common face or region in the drawing, so must those they map to in the target. Also, there are two additional constraints: (1) perpendicular intersections map to perpendicular intersections, but non-perpendicular intersections map to either perpendicular or non-perpendicular intersections (so for instance a non-right triangle in the source can map to a right triangle in the target but not the other way around), and (2) lines only map to lines, circles to circles, and arcs to arcs. The outline of the algorithm is as follows:

1. Apply each composite shape, matching it to the target drawing as many times as possible using backtracking constraint satisfaction with the composite shape elements as variables, target intersection graph elements as values, and matching graph structure as constraints.
 - Group symmetric composite shape mappings
2. For each composite shape mapping, break it into its basic shape mappings
 - Group symmetric basic shape mappings
3. For each set of symmetric basic and composite shape mappings, instantiate a new shape in the target drawing
4. Return a mapping from the source shapes to the target shapes that each one instantiated

Note that, in general, this shape-level mapping will *not be one-to-one*, and this is part of the idea for mapping from figure 1(a) to figure 1(c) could be found.

Transfer of Structural Elements

Once Archytas has a mapping between the basic and composite shapes of the source drawing and newly instantiated shapes of the target drawing, it needs to transfer the structural model from the source to the target. From these shape-level mappings we can hypothesize that if two shapes match then they should therefore depict the same component, and likewise for composite shapes and connections. The steps are to begin with the mapped shapes, and transfer the components and connections depicted by those mapped shapes, reconstructing the model iteratively.

As input to the structure transfer process, Archytas has a set of shape-level maps, basic shape maps and composite shape maps, each one associating a source shape with a newly instantiated target shape. From each one we can hypothesize a structural element, along with all its properties and variable parameters. The only difficulty is that some components are undepicted. For instance, in figure 1, neither figure depicts the crankcase. Archytas transfers undepicted components based on their connections to other components actually shown in the drawing. The output of this process is a new structural model of the target and a mapping from the source to the target that, once again, may not be one-to-one.

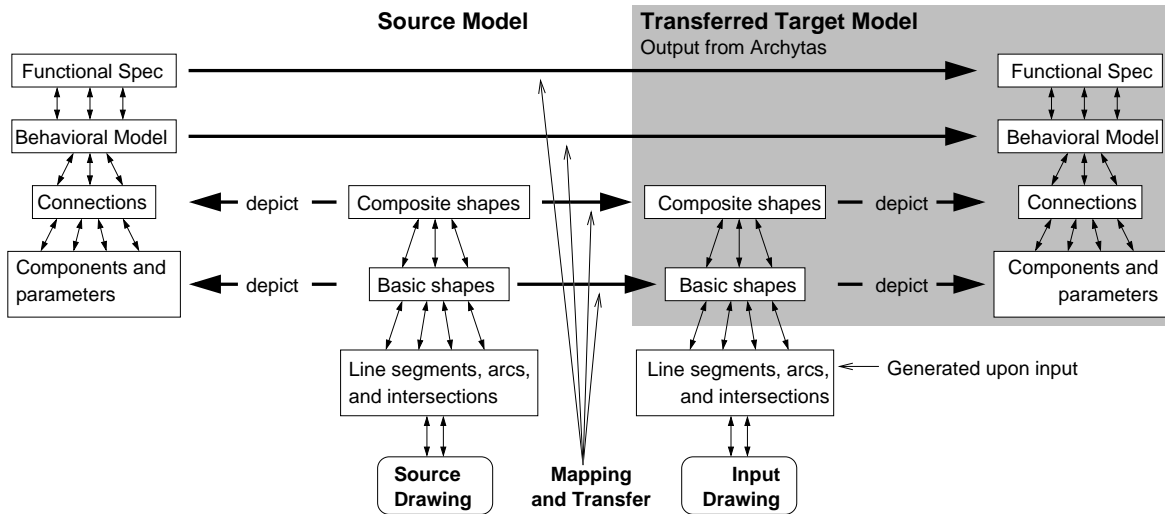


Figure 7: The multi-level hierarchy shown in figure 3 is transferred onto the target by first generating the line intersection graph of the input drawing, and then matching first composite shapes and then, when two composites overlap appropriately, basic shapes. Each basic and composite shape implies a component and structural relation, respectively, thereby suggesting a hypothetical structural model of the target. This component-level mapping is then used to transfer behavior and function.

Transfer of Behavior and Function

With a structural mapping in place, Archytas now needs to transfer behavior. Each behavior is a sequence of qualitative states of one of the components, and so the structural mapping directly informs the behavioral mapping to be constructed. However, there is a major complication: transitions in SBF can be annotated by a number of causes: conditions under which the transition may take place, for instance, other transitions which trigger this one, states of other components which must hold, important structural relations which allow the transition to occur, parametric equations, primitive functions, and so on. Since the structural mapping may not be one-to-one, the behavioral mapping may not be, either. For instance, if a given transition in one behavior is caused by another transition in another behavior, and the mapping of behaviors to behaviors (tracking that of components) is not one-to-one, then which behavior is it linked to?

Take, for example, the single piston crankshaft source with the double piston crankshaft target, figures 1(a) and (c). The downwards motion of the piston is caused by an external stimulus, but the upwards motion is caused by the continuing rotation of the crankshaft through the connecting rod. Thus, the transition of downward motion to upward motion occurs under the condition that the connecting rod so moves (which in turn occurs under condition that the crankshaft continues to turn, which it does so according to physical principles). But there are two pistons in the target, so for each piston, which state transition, which rod's behavior, causes its own? The answer is clear: the behavior of the rod connected to that piston. Thus, for each component, Archytas computes a *sphere of influence*, which is the set of other components that component is connected to, those connections themselves, and all the properties, parameters, and primitive functions of all of the components. When a behavior reference any of these elements, it is assumed to be within that sphere of influence.

Since the representation of function is simpler in SBF, the transfer of function is straightforward. A functional specification consists of an input and an output state, and a link to a behavior by which the function is realized. Since these states are behavioral states, the transfer of function is simply a lookup in the behavioral mapping.

Results

Archytas has been implemented in Common Lisp. There were 26 test drawings for Archytas (including source drawings) across three domains: that of the piston and crankshaft examples (figure 1), that of the door latch examples (figure 2), and that of pulley systems (not shown). These drawings representing a range of kinds of differences between source and target. The results are summarized in table 1.

Two interesting results are for the targets in figure 1(c) (the two piston and single crankshaft example) with 1(a) and its model as the source, and that of figure 2(c) (the double door latch) with 2(a) as its source. In both cases, the “correct” model for each target had two of some components and one of others. In the piston and crankshaft example this worked correctly, as each individual basic and composite shape pattern fit and overlapped just as it should have, producing the correct model. However, in the door latch example, the door shape—the two rectangles above and below the bolt—have changed. Now there are two of them surrounded by a rectangle (left side of the drawing and the right side) instead of just one, and this changes the shape, so that the door shape cannot match the drawing. Since this component was critical to the model, the transfer of the model failed. In general, the method is robust when the shape patterns do not change structurally—e.g. by gaining in complexity—but only in their configuration. As new drawings may contain unknown shape patterns, we would like the method to be able to acquire these new patterns, but this remains future work.

Difference	PC	DL	PL
Device State	yes	yes	yes
Dimension	yes	yes	yes
Orientation	yes	yes	yes
Perspective	no	no	no
Component Shapes	no	no	no
Number of Components	yes	no	no

Table 1: A brief summary of the differences between the source and target drawings that Archytas can and cannot handle: PC refers to differences tested with the piston and crankshaft example, DL refers to the door latch example, and PL refers to a third domain, that of pulleys. The categories are differences in the state of the device, the dimensions of some of the components, the orientation of the drawing in 2 dimensions, the perspective of the drawing in 3 dimensions, differences in the shapes of components, and differences in the number of components.

Related Work

GeoRep (Ferguson & Forbus, 2000) is a diagrammatic reasoning system which takes an arbitrary 2D line drawing as input and gives as output a description of the physical system depicted in the drawing. GeoRep is organized as a two-stage forward-chaining reasoner, in which a line drawing passes through a domain-independent low-level relational describer (LLRD) that recognizes lines and polygons, and from there a high-level relational describer (HLRD) that applies a domain-specific rule set to produce a final representation of the system in the diagram. Applications of applications of GeoRep typically derive structure from shape in HLRD; some applications have also used structure-mapping (Falkenhainer, Forbus, & Gentner, 1990) for making further inferences by analogy largely at the level of structure. By contrast, our work infers both shape and structure by analogy.

Recently Klenk et al. (Klenk, Forbus, Tomai, Kim, & Kyckelhahn, 2005) have developed a system for answering questions about simple mechanical systems using sketches. Their system has an user draw glyphs as separate entities and provide conceptual labels for these glyphs, so that the system need only compute the low-level visual relationships between the glyphs. It then uses structure mapping for making candidate inferences to complete the user-supplied model for use by a qualitative physical reasoner to answer questions about the physical system. Unlike Archytas, Klenk et. al.'s system does not recognize sketches or drawings. Further, their system does not use analogy to infer the entire model, but only to make inferences about an incomplete model given by the user, and to aid in answering a question.

Conclusions

In this paper, we considered the task of acquisition of a teleological model of a kinematics device from a 2D line drawing of that device by constructing its teleological model. Our method of compositional analogy first constructs a representation of the lines and the intersections in the target drawing, then uses mappings at the level of line intersections to transfer shape patterns from the source case to the target, next uses

the mappings at the level of shapes to transfer the structural model of the device, and finally uses mappings at the structural level to transfer the behavioral and functional representations of the device. We have shown that by relaxing the constraints of one-to-one mapping and by interleaving the mapping and transfer processes, a model can be inferred of the device depicted in a drawing by analogy without any pre-defined vocabulary of shapes beyond those that are actually present in a known drawing.

Analogy-based comprehension is often presented as the problem of analogical mapping (Falkenhainer et al., 1990; Holyoak & Thagard, 1989), and specifically that of structural alignment of *existing* representations. As a result, the tendency is to regard *an* analogy as *an* alignment or mapping, even when multiple mappings are used. Mappings at multiple levels of abstraction and aggregation requires an expansion of this view. For instance a whole mapping at one level can become a single “match hypothesis” at the next level. The role of mappings with respect to transfer thus changes when we consider analogy as a mechanism for *constructing* models at multiple levels of abstraction.

Acknowledgments

This research has been supported in part by an NSF (IIS) grant (Award number 0534266) on Multimodal Case-Based Reasoning in Modeling and Design.

References

- Falkenhainer, B., Forbus, K. D., & Gentner, D. (1990). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41, 1–63.
- Ferguson, R. W., & Forbus, K. D. (2000). GeoRep: A flexible tool for spatial representation of line drawings. In *Proc. 17th National Conf. on Artificial Intelligence (AAAI-2000)*. Austin, Texas: AAAI Press.
- Goel, A. K. (1991). Model revision: A theory of incremental model learning. In *Proc. 8th International Conf. on Machine Learning (ICML-91)* (pp. 605–609). San Mateo, CA: Morgan Kaufmann.
- Goel, A. K., & Chandrasekaran, B. (1989). Functional representation in design and redesign problem solving. In *Proc. 11th International Joint Conf. on Artificial Intelligence (IJCAI-89)* (p. 1388-1394). San Mateo, CA: Morgan Kaufmann.
- Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3), 295–355.
- Klenk, M., Forbus, K. D., Tomai, E., Kim, H., & Kyckelhahn, B. (2005). Solving everyday physical reasoning problems by analogy using sketches. In *Proc. 20th National Conf. on Artificial Intelligence (AAAI-05)* (pp. 209–215). Menlo Park, CA: AAAI Press.
- Yaner, P. W., & Goel, A. K. (2007). Understanding drawings by compositional analogy. In *Proc. 20th International Joint Conf. on Artificial Intelligence (IJCAI-07)* (pp. 1131–1137). San Mateo, CA: Morgan Kaufmann.